

(2) Japanese Patent Application Laid-Open No. 4-328665 (1992)

“Multiprocessor System and Interruption Arbitration System Thereof”

The following is an extract relevant to the present application.

5

This invention relates to a close-coupling type multiprocessor system, and more particularly, it is configured to allow a distribution of I/O interruptions to each processor in accordance with the load status of said each processor without concentrating on one processor.

10

Priority control and round robin control are used in combination, and each processor is provided with two kinds of priorities and the highest-priority processor is determined in accordance with combinations of two kinds of priorities.

(51)Int.Cl.⁵

G 0 6 F 13/362
9/46
13/24
15/16

識別記号

5 1 0 A 7052-5B
3 2 2 C 8120-5B
3 1 0 Z 7052-5B
3 1 0 P 8840-5L

庁内整理番号

F I

技術表示箇所

審査請求 有 請求項の数12(全 7 頁)

(21)出願番号

特願平3-116694

(22)出願日

平成3年(1991)4月22日

(71)出願人 390009531

インターナショナル・ビジネス・マシー
ズ・コーポレーション

INTERNATIONAL BUSIN
ESS MASCHINES CORPO
RATION

アメリカ合衆国10504、ニューヨーク州

アーモンク (番地なし)

(72)発明者 福田 宗弘

神奈川県相模原市相模大野3-7-6

(74)代理人 弁理士 頓宮 孝一 (外4名)

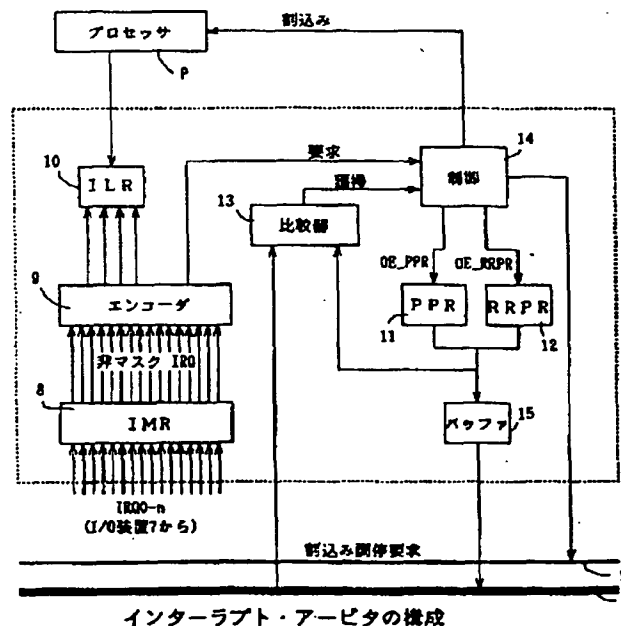
最終頁に続く

(54)【発明の名称】 マルチプロセッサ・システムおよびその割り込み調停装置

(57)【要約】

【目的】密結合型マルチプロセッサ・システムにおいて、プロセッサPの負荷状況に応じて、しかも1つのプロセッサPに偏ることなく、I/O割り込みを各プロセッサPに分配する。

【構成】各プロセッサPに設けられた割り込み調停回路3がI/O装置から割り込み要求を受取り、プロセッサPの負荷状況を表すパラメータを第1の割り込み優先度PPRとして割り込みの調停を行い、この調停で1個のプロセッサPが決まらないときに、さらに付加的に循環的に変化する第2の割り込み優先度RRPRにしたがって最終的に1個のプロセッサPを選択する。



【特許請求の範囲】

【請求項 1】複数のプロセッサと、上記プロセッサの各々に対応して設けられ、上記プロセッサの各々について第 1 の割込み優先度を示すデータを保持する第 1 の割込み優先度保持手段と、上記プロセッサの各々に対応して設けられ、上記プロセッサの各々について第 2 の割込み優先度を示すデータを保持する第 2 の割込み優先度保持手段であって、上記プロセッサごとに第 2 の割込み優先度を異ならせ、かつ所定のイベントの発生のたびに上記第 2 の優先度を変更するようにしたものと、上記プロセッサの各々に対応して設けられ、当該プロセッサを含む複数のプロセッサに対して割込み要求があり、当該プロセッサの第 1 の割込み優先度のみが最優先のときに、上記割込み要求を受付ける手段と、上記プロセッサの各々に対応して設けられ、当該プロセッサを含む複数のプロセッサに対して割込み要求があり、当該プロセッサの第 1 の割込み優先度を含む複数の第 1 の割込み優先度が最優先であり、さらに当該プロセッサの第 2 の割込み優先度が、上記最優先の第 1 の優先度を保持する他のプロセッサの第 2 の割込み優先度より優先のときに、上記割込みを受付ける手段とを有することを特徴とするマルチプロセッサ・システム。

【請求項 2】上記第 2 の割込み優先度がすべてのプロセッサにおいて昇順に変化し、上限値で下限値へ復帰する請求項 1 記載のマルチプロセッサ・システム。

【請求項 3】上記第 2 の割込み優先度がすべてのプロセッサにおいて降順に変化し、下限値で上限値へ復帰する請求項 1 記載のマルチプロセッサ・システム。

【請求項 4】上記所定のイベントが上記割込み要求に対する上記プロセッサの選択動作である請求項 1、2 または 3 記載のマルチプロセッサ・システム。

【請求項 5】上記プロセッサの各々の第 1 の割込み優先度は当該プロセッサで実行されているプロセスの優先度が高くなればなるほど低くなるようにした請求項 1、2、3 または 4 記載のマルチプロセッサ・システム。

【請求項 6】上記プロセッサの各々の第 1 の割込み優先度は当該プロセッサの割込み要求待ち行列中の割込み要求の数が多くなればなるほど低くなるようにした請求項 1、2、3 または 4 記載のマルチプロセッサ・システム。

【請求項 7】複数のプロセッサと、上記プロセッサの各々に対応して設けられ、上記プロセッサの各々について第 1 の割込み優先度を示すデータを保持する第 1 の割込み優先度保持手段と、上記プロセッサの各々に対応して設けられ、上記プロセッサの各々について第 2 の割込み優先度を示すデータを保持する第 2 の割込み優先度保持手段であって、設定可能な上限値および設定可能な下限値の間において、所定のイベントの発生のたびに上記第 2 の優先度を変更するようにしたものと、上記プロセッサの各々に対応して設けられ、当該プロセッサを含む複

数のプロセッサに対して割込み要求があり、当該プロセッサの第 1 の割込み優先度のみが最優先のときに、上記割込み要求を受付ける手段と、上記プロセッサの各々に対応して設けられ、当該プロセッサを含む複数のプロセッサに対して割込み要求があり、当該プロセッサの第 1 の割込み優先度を含む複数の第 1 の割込み優先度が最優先であり、さらに当該プロセッサの第 2 の割込み優先度が、上記最優先の第 1 の優先度を保持する他のプロセッサの第 2 の割込み優先度より優先のときに、上記割込みを受付ける手段とを有することを特徴とするマルチプロセッサ・システム。

【請求項 8】上記上限値および上記下限値が同一値をとるプロセッサでは上記第 2 の割込み優先度が上記同一値に固定される請求項 7 記載のマルチプロセッサ・システム。

【請求項 9】複数のプロセッサと、上記プロセッサの各々に対応して設けられ、上記プロセッサの各々について第 1 の割込み優先度を示すデータを保持する第 1 の割込み優先度保持手段と、上記プロセッサの各々に対応して設けられ、上記プロセッサの各々について第 2 の割込み優先度を示すデータを保持する第 2 の割込み優先度保持手段であって、上記プロセッサごとに第 2 の割込み優先度を異ならせ、かつ所定のイベントの発生のたびに上記第 2 の優先度を変更するようにしたものと、上記プロセッサの各々に対応して設けられ、当該プロセッサを含む複数のプロセッサに対する割込み要求に対して、当該プロセッサを含む複数のプロセッサの第 1 の割込み優先度に基づいて、当該プロセッサのみが選択されるときは上記割込み要求を当該プロセッサに受付けさせる手段と、上記プロセッサの各々に対応して設けられ、当該プロセッサを含む複数のプロセッサに対する割込み要求に対して、当該プロセッサを含む複数のプロセッサの第 1 の割込み優先度に基づいて当該プロセッサを含む複数のプロセッサが選択され、かつ選択された複数のプロセッサの第 2 の割込み優先度に基づいて、当該プロセッサが選択できるときは上記割込み要求を当該プロセッサに受付けさせる手段とを有することを特徴とするマルチプロセッサ・システム。

【請求項 10】複数のプロセッサと、調停バスと、上記プロセッサの各々に対応して設けられ、かつ上記調停バスに結合され、当該プロセッサを含む複数のプロセッサがバス・アクセスを要求するときに、当該プロセッサがバス・アクセスを優先的に保持するかどうかを決定するバス調停手段と、上記プロセッサの各々に対応して設けられ、上記プロセッサの各々について第 1 の割込み優先度を示すデータを保持する第 1 の割込み優先度保持手段と、上記プロセッサの各々に対応して設けられ、上記プロセッサの各々について第 2 の割込み優先度を示すデータを保持する第 2 の割込み優先度保持手段であって、上記プロセッサごとに第 2 の割込み優先度を異ならせ、か

つ所定のイベントの発生のたびに上記第2の優先度を変更するようにしたものと、上記プロセッサの各々に対応して設けられ、かつ上記調停バスに結合され、当該プロセッサを含む複数のプロセッサに対する割り込み要求に対して、当該プロセッサを含む複数のプロセッサの第1の割り込み優先度に基づいて、当該プロセッサのみが選択されるときは上記割り込み要求を当該プロセッサに受けさせる第1の割り込み調停手段と、上記プロセッサの各々に対応して設けられ、かつ上記調停バスに結合され、当該プロセッサを含む複数のプロセッサに対する割り込み要求に対して、当該プロセッサを含む複数のプロセッサの第1の割り込み優先度に基づいて当該プロセッサを含む複数のプロセッサが選択され、かつ選択された複数のプロセッサの第2の割り込み優先度に基づいて、当該プロセッサが選択できるときは上記割り込み要求を当該プロセッサに受けさせる第2の割り込み調停手段と上記割り込み要求に基づいて上記バス調停手段の調停バスの使用を禁止する手段とを有することを特徴とするマルチプロセッサ・システム。

【請求項11】第1の割り込み優先度を示すデータを保持する第1の割り込み優先度保持手段と、第2の割り込み優先度を示すデータを保持する第2の割り込み優先度保持手段であって、かつ所定のイベントの発生のたびに上記第2の優先度を変更するようにしたものと、第1の割り込み優先度に基づいて、割り込み要求を受け付けるかどうかを決定する手段と、上記第1の割り込み優先度に基づいて割り込み要求を受け付けるかどうかを決定できないときに、上記第2の割り込み優先度に基づいて、上記割り込み要求を受け付けるかどうかを決定する手段とを有することを特徴とするマルチプロセッサ・システムの割り込み調停装置。

【請求項12】上記第1の優先度はマルチプロセッサ中においてすべて異なる値をとるように初期設定できるようにされている請求項11記載のマルチプロセッサ・システムの割り込み調停装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、密結合型マルチプロセッサ・システムに関し、とくにプロセッサの負荷状況に応じて、しかも1つのプロセッサに偏ることなく、I/O割り込みを各プロセッサに分配できるようにしたものである。

【0002】

【従来の技術】近年のハードウェア技術の向上により、多数のプロセッサを搭載した密結合型マルチプロセッサ・ワークステーションの実用化が進んでいる。この種のワークステーションでは、高速な数値演算機能に加えて、多種多様なI/O機能を提供することが求められている。

【0003】従来システムの多くは、各プロセッサに対して局所的にI/O装置を接続する形態、もしくは、特定

のプロセッサがI/O装置を集中管理する形態（たとえば特開昭63-147252号）をとる。このようなシステムは、特定のプロセッサが特定のI/Oを制御することを前提としている。したがって、その実現が容易である反面、I/O処理の負荷分散は極めて不可能に近く、高度なI/O機能を効率良くユーザに提供することは難しい。

【0004】これに対して、I/O処理の負荷分散を密結合型マルチプロセッサ・システム上で行う試みが行なわれつつある。I/O処理の負荷分散を実現するためには、システム中の全てのプロセッサが、平等にかつ直接、I/O装置を操作できることが必須である。特に、I/O装置からの割り込みを各プロセッサに分配する機能を実現する必要がある。しかしながら、この割り込みの分配に際して、1つの割り込みを同時に2台以上のプロセッサに受け取らないように割り込みを調停しなければならない。この割り込み調停が実現できないと、1つのI/O装置からの要求が重複して処理されることになる。また、最も負荷の軽いジョブ、あるいは、優先度の低いジョブを実行しているプロセッサへ割り込みを送信することが、I/Oの負荷分散を実現する上で重要である。

【0005】I/O処理の負荷分散を行っているシステムについては、"The Design and Development of a Very High Speed System Bus - The Encore Multimax Nanobus", D.J.Schanin, Proceedings of Fall Joint Computer Conference, pp.410-418、1986年、11月、および、"Balance: A Shared Memory Multiprocessor System", S.S.Thakkar, P.Gifford, G.Fielland, Proceedings of 2nd International conference on Supercomputing, pp.93-101、1987年5月に開示がある。これら2つのシステムは、割り込み調停を行うための機能を備えている。これら2つのシステムでは、各プロセッサの実行しているプロセスの優先度、もしくは、割り込み待ち行列にある未処理の割り込みの数をパラメータとして割り込み調停を行い、I/Oの負荷分散を実現しようとしている。しかし、2台以上のプロセッサが、調停に勝つ可能性がある。そこで、プロセッサ固有のスロット番号によって、割り込みを受け付けるべきプロセッサを最終的に1台選択する。

【0006】しかしながら、この2つの割り込み調停方式は、I/Oの負荷分散を十分に実現することができない。なぜなら、全てのプロセッサが同一のプライオリティ・レベルで処理を行っているとき、常に物理的位置を示すスロット番号によって決定されたプロセッサへ割り込みが送られることになる。また、どのプロセッサの割り込み待ち行列も空である場合には、たとえ重要なジョブを実行しようとも常に決まったプロセッサへ割り込みが送られる。また、割り込みレベルごとに割り込み可能なプロセッサ群を指定することは、全く不可能か、あるいは、システムの起動時にのみ設定可能である。したがって、システムの動作中に各プロセッサの負荷状況に応じて割り込み

可能なプロセッサ群を動的に指定し、変更することはできない。さらに、割込み調停を行うために、その制御ハードウェアをプロセッサにのみ装備した例は実在しない。現存のシステムでは、プロセッサのみならず、I/O装置に制御ロジックを加えるか、もしくは、特別なI/O装置を新設するとともに、割込み調停用の通信線をシステム・バスに付加している。したがって、割込み調停を実現するための制御ハードウェアの規模も無視できない。調停の実現方法は、メッセージ通信を繰り返して行うもの、非同期型バック・オフ方式で行うもの等、様々であるが、短時間で割込みを調停し、プロセッサに送信することは期待できない。

【0007】この発明と関連する他の先行技術には特開昭61-87448号公報、特開昭62-187955号および特開平1-279354号公報がある。特開昭61-87448号はバス調停のたびにバス調停用の優先度を変更することを開示している。特開昭62-187955号公報はバス調停のたびにバス調停用の優先度を循環的に変更することを開示している。また特開平1-279354号公報はバス調停のたびにバス調停用の優先度を循環的に変更させ、さらに上限値および下限値を各プロセッサごとに設定可能とし、各プロセッサではその上限値および下限値の間で優先度が変化するようにすることを開示している。この場合所定のプロセッサの上限値および下限値を同一にすればその優先度をその値に固定できる。しかしながら以上の先行技術はすべてバス要求の調停であり、しかもこれらには一回目の調停で割込みの負荷分散を行い、しかも付加的な二回目の調停で割込みの偏りをなくすという点についてはなんら教示がない。

【0008】

【発明が解決しようとする課題】この発明は、以上の事情を考慮してなされたものであり、プロセッサの負荷状況に応じて、しかも1つのプロセッサに偏ることなく、I/O割込みを各プロセッサに分配できるようにしたマルチプロセッサ・システムのI/O割込みを提供することを目的としている。さらにこの発明はI/O割込みの制御を既存のマルチプロセッサ・システムに容易に適用できるようにすることを目的としている。

【0009】

【課題を解決するための手段】この発明では以上の目的を達成するために、プロセッサの負荷状況を表すパラメータを第1の優先度としてI/O割込みの調停を行い、この調停で1個のプロセッサが決まらないときに、循環的に変化する第2の優先度にしたがって最終的に1個のプロセッサを選択できるようにしている。またこの発明ではプロセッサからのバス要求の調停を行うバス・アービタをI/O割込み用に兼用し既存のマルチプロセッサ・システムへの適用を容易にしている。すなわち従来の密結合型マルチプロセッサ・システムでは、各プロセッサに

装備されたバス・アービタが、調停線を介して相互に通信し、プロセッサからのバス使用要求を調停するものが多い。本発明は、この調停線を割込みの調停に流用し、各プロセッサに割込みの調停を行う回路を付加することにより、極めて簡単に実現できる。回路は、従来の割込みコントローラのロジックに若干のレジスタと比較器を加えたものである。この回路を、バス・アービタと区別するためにインタラプト・アービタと呼ぶことにする。

【0010】

【実施例】以下この発明の実施例について図面を参照して説明しよう。図1はこの発明が適用されたマルチプロセッサ・システムの割込み調停機構を全体的に示すものである。図1においてプロセッサP1…PNは共有バス1（図ではデータバス、アドレスバス、制御バスは省略している）に共通に結合されている。各プロセッサP（このプロセッサを指し示す場合以外はサフィックスを除いて示す）はバス・アービタ2およびインタラプト・アービタ3を具備し、バス・アービタ2でプロセッサPからのバス要求を調停し、インタラプト・アービタ3でI/O割込み要求を調停する。バス・アービタ2およびインタラプト・アービタ3の双方は調停線4および割込み調停要求線5に共通に接続され、インタラプト・アービタ3はさらに割込み要求線6に共通に接続されている。この割込み要求線6にはI/O装置7が接続されている。

【0011】インタラプト・アービタ3は、I/O装置7から割込み要求を割込み要求線6を介して直接入力する。ある割込み要求がI/O装置7から発行されると、各インタラプト・アービタ3は、割込み調停要求線5をLOWに落とし、バス・アービタ2に対して調停線4の使用を一時禁止する。割込み調停要求線5は、ワイヤード・オアでアクティブLOWの信号線である。インタラプト・アービタ3は、前のメモリ・バス・サイクルが終了すると、この調停線4を介して相互に通信し、割込みの調停を行う。

【0012】割込み調停は、図2を参照して説明するように、最初に各プロセッサPが実行しているプロセスの優先度を用いて行われる。プロセスの優先度はプロセスの種類たとえばOSカーネル、アプリケーション等に応じて割り当てられる。最低のプロセスの実行優先度を持つプロセッサPが勝ち残る。プロセスの優先度が同じプロセッサPが2台以上存在する場合があるので、ラウンド・ロビン方式を使用して第2回目の調停を行う。ラウンド・ロビンは、予め各アービタに固有の優先度を与え、調停を行う毎に各アービタがその優先度をローテイトし、優先度の最も高いプロセッサを選択する方式である。このローテイトにより、常に公平な調停を行うことができる。この2回にわたる調停により、インタラプト・アービタ3は、I/O割込みを受け取るプロセッサPを唯一つ決定し、そのプロセッサPに割込み信号をかけ

る。

【0013】図2はインタラプト・アービタの実現例を示す。図2において、インタラプト・アービタ3は、割込みマスク・レジスタ(IMR)8、エンコーダ9、割込みレベル・レジスタ(ILR)10、プロセス・プライオリティ・レジスタ(PPR)11、ラウンド・ロビン・プライオリティ・レジスタ(RRPR)12、コンパレータ13および制御ロジック14等から構成される。このうち、IMR8、エンコーダ9、ILR10は、従来の割込みコントローラが提供する機能である。IMR8は、マスクすべき割込み要求レベルを指定する。エンコーダ9は、IMR8を通過した割込み要求レベル(IRQ)、すなわち、マスクされなかった割込み要求レベル(マスクされない IRQ)のうち、最もプライオリティの高い割込み要求レベル(IRQ)を1つ選択する組合せロジックである。ILR10は、割込みに応答したプロセッサに対して、エンコーダ2で選択した割込み要求レベルを返すレジスタである。

【0014】したがって、本発明を実現する上で新たに必要となる構成要素は、PPR11とRRPR12の2つのレジスタ、コンパレータ13、および、その制御ロジック14だけである。PPR11およびRRPR12のデータはバッファ15に一時記憶される。PPR11は、プロセスの実行優先度に応じたI/O割込みの優先度を指定し(プロセスの実行優先度が高いほどI/O割込みの優先度が低くなる)、プロセスの実行優先度による第1回目の調停に使用される。各インタラプト・アービタ3のRRPR12は、プロセッサの個数の範囲で循環計数するカウンタであり、システム起動時に固有の優先度に初期化され、各インタラプト・アービタごとに異なる値を有している。カウンタは割込み調停動作、すなわち割込み調停パルス(図3、このパルスは割込み調停要求の信号がLOWでかつBUS_BusyがHIGHのときにHIGHになる信号である)をカウントし、その内容は、割込み調停ごとにローテイトする。すなわち0の優先度が1に、1の優先度が2に、3の優先度が4に、…、(N-1)

(Nはプロセッサの個数)の優先度が0に変化する。あるいはに(N-1)の優先度が(N-2)に、(N-2)の優先度が(N-2)に、…、0の優先度が(N-1)に変化する。RRPR12の内容は、ラウンド・ロビン方式による第2回目の調停に使用される。エンコーダ9は、選択した割込み要求レベルが変化すると要求信号を制御ロジック14に送る。制御ロジック14は、この要求信号によって起動される。この回路14は、割込み調停要求線5をLOWに落とし、バス・アービタ2に対して調停線4の使用を一時禁止する。前のメモリ・バス・サイクルが終了すると、プロセスの実行優先度による第1回目の割込み調停を行うために、OE_PPR(Output Enable PPR)信号により、PPR11の内容を調停線4に送出する。第1回目の調停に勝ち残ると、ラウンド・ロ

ビン方式による第2回目の調停を行うために、OE_RRPR(Output Enable RRPR)信号により、RRPR12の内容を調停線4に送出する。コンパレータ13は、第1回目の割込み調停では、PPR11の内容と調停線4に送出されている他のインタラプト・アービタ3の優先度を比較する。自分のプロセスの実行優先度が最低であるとき、制御ロジックに獲得信号を送って、第1回目の調停に勝ち残ったことを通知する。第2回目の調停では、RRPR12の内容と他の優先度を比較する。自分のRRPR12の優先度が最高であるとき、制御ロジック14に獲得信号を送って、第2回目の調停で選択されたことを通知する。制御ロジック14は、これを受けてプロセッサに対して割込み信号を送出し、割込みをかける。

【0015】図3は、インタラプト・アービタ3による割込み調停のタイミング・チャートを示している。この例では、共有バス1がクロック同期であるものとする。この図において割込み要求(i)は、1からnレベルの割込み要求線のうち、i番目の割込み要求線であることを示す。バス・ビジー#信号は、アクティブLOWのトラステート、または、ワイヤード・オア信号線で、この信号がLOWであるとき、共有バス1がメモリ・サイクル中であることを示す。したがって、バス1の使用要求の調停、または、割込みの調停は、このバス・ビジー#信号がHIGHであるときに行われる。割込み信号は、各インタラプト・アービタ3がプロセッサPへ割込みを通知するための信号である。図3の割込み信号は、割込み調停で選択された唯一のインタラプト・アービタ3の割込み信号を示す。以下に図3における割込み調停の実施手順の一例を説明する。

【0016】(1)バス・クロック1において、あるI/O装置が割込み要求(i)信号を立ち上げて割込み要求を発行する。各インタラプト・アービタ3では、この割込み要求(i)信号をIMR8でマスクしていない場合、エンコーダ9が最もレベルの高い割込み要求を選択し直す。エンコーダ9の値が変化した場合には、要求信号が制御ロジック14に送られる。

(2)バス・クロック2において、インタラプト・アービタ3の制御ロジック14が起動し、割込み調停要求#信号を共有バス1に送出する。これにより、次のアービトレーション・サイクルは割込み調停だけに有効となる。したがって、バス・アービタ2は、割込み調停要求#信号がLOWになると次のアービトレーション・サイクルの参入を見合わせる。

(3)バス・クロック3において、バス・ビジー#信号がHIGHになり、メモリ・バス・サイクルが終了し、アービトレーション・サイクルに入る。割込み調停要求#信号がLOWであるので、バス・アービタ2は、バス使用要求の調停を行うことはできない。このサイクルでは、1台以上のインタラプト・アービタ3が割込み調停に参入する。割込み調停は、2サイクルで行なわれる。バス・

クロック3で、インタラプト・アービタ3は、PPR11の値を調停線4に出力し、各々のプロセッサPが実行しているプロセスの優先度で割り込み調停を行う。最も低いプロセスの実行優先度を持つインタラプト・アービタ3が勝ち残る。

(4) バス・クロック4において、最低のプロセス実行優先度を持つインタラプト・アービタ3が、RRPR12の値を調停線4に出力し、ラウンド・ロビンによる調停を行う。RRPR12の内容が最高であるインタラプト・アービタ3が唯一選択される。

(5) バス・クロック5において、選択されたインタラプト・アービタ3は、割り込み信号により、プロセッサPに対して割り込みをかける。割り込まれたプロセッサPは、ILR10を読むことによって受け付けた割り込みのレベルを知ることができる。割り込み調停に参入していたインタラプト・アービタ3は、割り込み調停要求#信号をHIGHに上げてバス・アービタ3に対して、調停線4の使用を許可する。バス・アービタ2は、バス・クロック5において、割り込み以外の通常のバス使用要求の調停を行うことができる。

(6) バス・クロック6において、バス使用要求の調停に勝ったプロセッサPがバス・ビジー#信号をLOWに落とし、メモリ・バス・サイクルを開始する。

【0017】I/O装置からの割り込みは、最も負荷の軽い、あるいは、優先度の低いプロセスを実行しているプロセッサに均一に送ることが望ましい。この実施例では、ソフトウェアによる設定が可能なPPR11の値を用いて、最も優先度の低いプロセスを実行しているプロセッサPを選ぶ。さらに、最低の実行優先度を持つプロセッサPが、2台以上存在する場合でも、常に決まったプロセッサPを選択するのではなく、ラウンド・ロビンによる調停により、その中から公平にプロセッサPを1台選択する。したがって、I/O処理の負荷分散を徹底することができる。

【0018】この実施例による割り込み調停は通常のシステム・バスの調停サイクルを利用して行う。インタラプト・アービタ3は、このサイクルでプロセスの実行優先度による調停を行ない、さらに、もう1サイクル使用して、ラウンド・ロビンによる調停を行う。したがって、通常のバス調停時間の高々、2倍の時間で割り込みを調停し、分配できる。集中して種々のレベルの割り込みが発生した場合でも、短期間に各プロセッサPに割り込みを分配できる。

【0019】1台のプロセッサPでのみOSカーネルが実行され、その他のプロセッサではユーザ・プロセスが実行される場合、種々のI/Oデバイス・ハンドラ、サーバが各プロセッサPに固定である場合には、あるレベルの割り込みを特定のプロセッサPへ送信できることが望ましい。また、すべてのプロセッサPで同一のOSカーネルを実行させたい場合、プロセッサ・グループごとに実

行すべきサーバが異なる場合には、あるレベルの割り込みを特定のプロセッサ・グループに送信できることが望ましい。本発明は、IMR8の変更により、レベルごとに割り込みを受け付けるプロセッサ、または、プロセッサ・グループをシステムの負荷状況に応じて動的に変更することができる。すなわち、割り込みレベルごとにプロセッサ群を個別に指定することができる。

【0020】また、この実施例は、インタラプト・アービタ3のロジック量が少なく、インタラプト・アービタ3を各プロセッサPに装備するだけで簡単に実現できる。また、I/O装置、コントローラ等に制御用のロジックを付加する必要がないので、既存のI/Oカードを使用することができる。割り込み調停に必要な通信線は、システム・バスの調停線を共有するので、バック・プレーンをほとんど変更する必要がない。したがって、若干のハードウェアの変更、および追加により、この実施例を既存のマルチプロセッサ・システムに極めて容易に適用できる。

【0021】なおこの発明は以上の実施例の詳細に限定されるものではなく、その趣旨を逸脱しない範囲で種々の変更が可能である。プロセスの優先度に応じて一回目の割り込み調停を行うかわりに、割り込み待ち行列中の割り込み要求の数が多いほど割り込みの優先度を低くするようにしてもよい。

【0022】また二回目の割り込み調停の優先度を完全なラウンド・ロビンにするのではなく、特開平1-279354号公報のように設定可能な上限値および下限値の間を変化する優先度を採用してもよい。この場合所定のプロセッサの上限値および下限値を所定の値にすれば、その優先度をその所定の値に固定できる。もちろんその他のプロセッサの優先度は割り込み調停に応じて変化する。

【0023】

【発明の効果】以上説明したようにこの発明によれば極めて簡単な構成で割り込みの調停を動的かつ高速に実現でき、既存のシステムへの適用が容易である。

【図面の簡単な説明】

【図1】この発明の実施例を全体として示すブロック図である。

【図2】図1の実施例のインタラプト・アービタの構成を示すブロック図である。

【図3】図2のインタラプト・アービタの操作を説明するためのタイミング・チャートである。

【符号の説明】P・・・プロセッサ

1・・・共有バス

2・・・バス・アービタ

3・・・インタラプト・アービタ

4・・・調停線

5・・・割り込み調停要求線

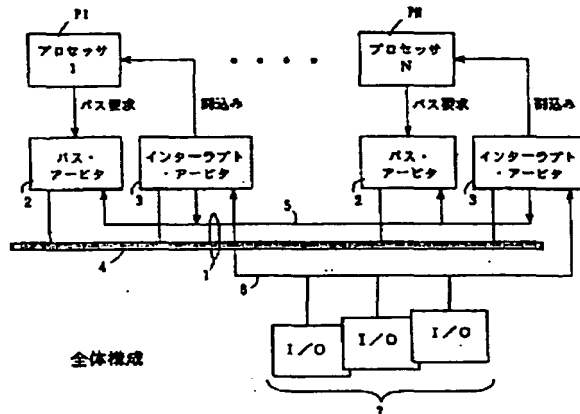
6・・・割り込み要求線

11・・・プロセス・プライオリティ・レジスタ

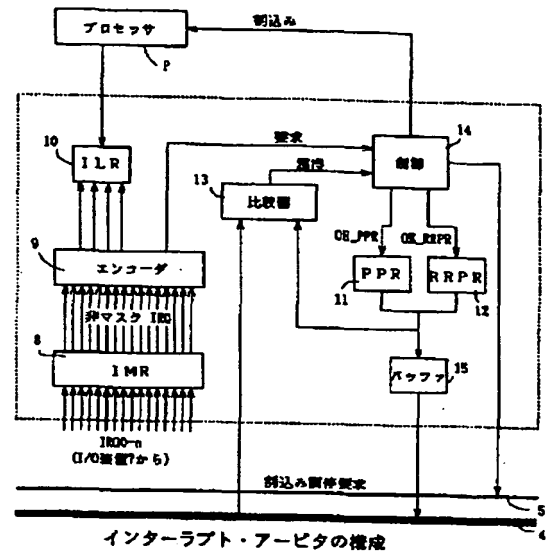
12・・・ラウンド・ロビン・プライオリティ・レジスタ
タ

13・・・コンパレータ
14・・・制御ロジック

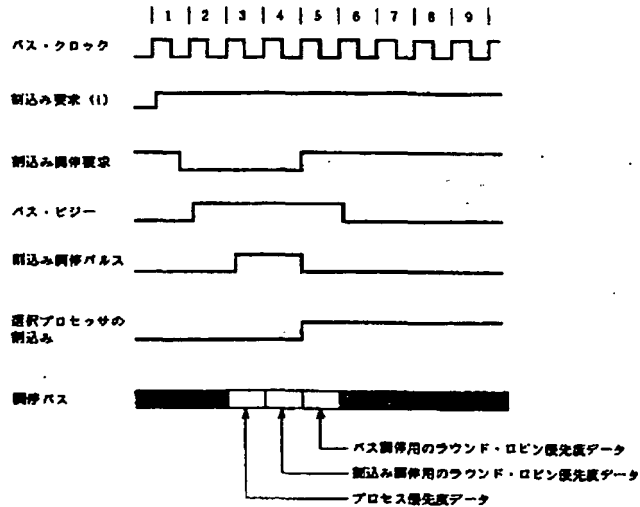
【図1】



【図2】



【図3】



フロントページの続き

(72)発明者 大庭 信之
神奈川県川崎市多摩区菅北浦 3-13-26-
404

(72)発明者 中田 武男
埼玉県川口市飯原町 3-3